INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **43**:717–728 (DOI: 10.1002/fld.615)

Unstructured Navier–Stokes grid generation at corners and ridges

Dmitri Sharov[1,✠], Hong Luo[1], Joseph D. Baum[1] and Rainald Löhner[2,*,†]

[1]*Science Applications International Corporation, 1710 SAIC Drive, MS 2-6-9, McLean, VA 22102, U.S.A.*
[2]*School of Computational Sciences, MS 4C7, George Mason University, Fairfax, VA 22030, U.S.A.*

SUMMARY

Problems related to automatic generation of highly stretched unstructured grids suitable for 3-D Reynolds-averaged Navier–Stokes computations are addressed. Special attention is given to treatment of such geometrical irregularities as convex and concave ridges as well as corners where the ridges meet. The existing unstructured grid generation approaches may fail or produce poor quality meshes in such geometrical regions. The proposed solution is based on special meshing of non-slip body surfaces resulting in smooth and robust volume meshing and high overall quality of generated grids. Several examples demonstrate the efficiency of the method for complex 3-D geometries. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: grid generation; unstructured grids; CFD; FEM

INTRODUCTION

Automatic unstructured grid generation for high-Reynolds viscous flows is one of the remaining CFD bottlenecks. Several solutions of the problem have been proposed recently [1–9]. However the existing approaches may experience reliability and grid quality problems in areas where the wetted surfaces feature sharp concave/convex ridges or corners.

The existing methods are based on combination of structured or semi-structured grids in viscous regions (close to the body), and unstructured, usually tetrahedral meshes elsewhere. The semi-structured meshes can be generated using prismatic mesh generators [1], which construct the prismatic elements by advancing the original surface triangulation in a direction normal to the body. The prismatic elements can be subdivided later into tetrahedra [4]. Another approach constructs the stretched tetrahedra directly using the advancing layers method [5]. In Reference [7] the viscous stretched mesh was generated between the non-slip surface

*Correspondence to: R. Löhner, School of Computational Sciences, M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, U.S.A.
†E-mail: rlohner@gmu.edu
✠Deceased.

Received 3 March 2002
Revised 20 June 2003

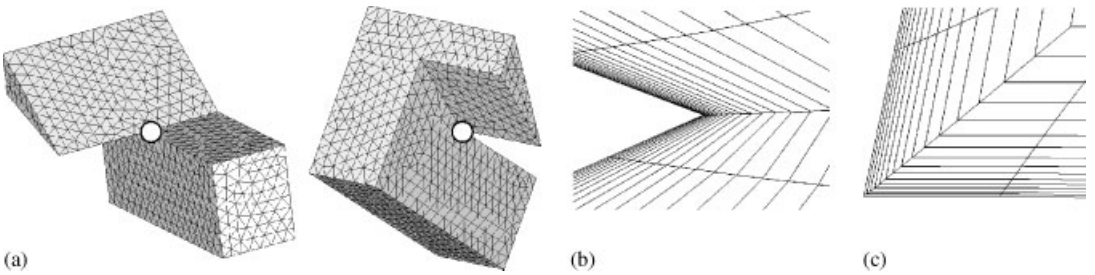

Figure 1. Problems associated with prismatic grid generation: (a) corner point where no prism can be generated; (b) convex ridge, where stretched elements are connected at sharp angles; (c) concave ridge, where stretched elements are connected at sharp angles and grid resolution inside the corner is not adequate.

and the Cartesian mesh. A special algorithm was designed to include corners and ridges into the mesh. Local node insertion with reconnection in viscous regions was proposed in Reference [8]. Yet another approach uses well-established structured mesh generators near the body surfaces and merges them via unstructured methodology [9]. Although the last approach was expected to generate better meshes in vicinity of the non-slip boundaries, it still has all the drawbacks of the conventional structured mesh generators, namely, user involvement in the generation process requiring complex user interface and high level of user expertise. The other existing approaches such as prismatic, advancing layers method, local insertion/reconnection method, and Cartesian method require less user input but in their original form still suffer from difficulties in meshing corners and ridges.

The first problem with the automatic semi-structured grid generators occurs at corners where several concave and convex ridges meet. Generation of prisms requires visibility of the newly created point from all the faces surrounding the original point. For some geometries such a point might not exist, and therefore the prismatic grid generator fails. An example of such a corner is given in Figure 1(a).

The second set of problems is associated with convex ridges such as at the trailing edge of a wing. Regular prismatic grid generators create only one normal direction from each point on the ridge, which results in extremely poor mesh quality. These meshes typically contain highly stretched elements connected to each other at sharp angles, see Figure 1(b). The problem can be treated by the introduction of multiple normal directions at the ridges [6]. Nevertheless, large discrepancies in neighbouring cell sizes will still occur. As a result, the mesh quality will be not adequate in the vicinity of the ridge.

Concave ridges such as at wing-fuselage junction (see Figure 1(c)) are also a serious source of grid generation problems. Even sophisticated elliptic or hyperbolic prismatic grid generators cannot guarantee adequate meshing there, resulting in meshes that usually lack appropriate required resolution.

The objective of the present effort is to develop an automatic method based on the prismatic grid generation technique that will generate better meshes close to corners and ridges. The proposed solution comes from understanding that the conventional isotropic triangulation of the non-slip body surfaces is not consistent with the topology of the viscous volume mesh. In fact, a stretched viscous mesh with high-cell aspect ratios is reasonable only for a flat plate or

near smooth surfaces. In the case of a singularity like a ridge or a corner, the stretched mesh should not be generated in the same way as for smooth surfaces. We propose to start the generation process from adequate surface meshing. Once the proper surface grid is generated, the volume meshing becomes a lot easier and more accurate. The proposed method makes it possible to mesh complex corner regions by non-stretched isotropic tetrahedra. For convex ridges the surface mesh clusters to the ridge and provides a high-quality viscous mesh without adding any extra normals. Clustering of the surface mesh to the concave ridges provides a consistent volume mesh.

SURFACE GRID GENERATION ALGORITHM

Surface mesh generation is one of the key elements of the proposed grid generation method. The surface meshing is designed to make the volume mesh generation more reliable as well as to improve the overall grid quality.

First, consider viscous mesh generation in the vicinity of corners. The most straightforward way here is to generate isotropic tetrahedral mesh at the corner, which will be gradually transferred to the stretched mesh away from the corner. This approach not only enhances robustness but also improves the mesh quality since stretched elements near the corner result in inaccurate flowfield predictions and stalling of convergence. Isotropic tetrahedra generation starting from the conventional surface triangulation leads to very large tetrahedra, which are not acceptable for viscous solution. Thus, the only reasonable solution is to refine the surface mesh at the corners so that triangles sharing the corner would be as small as the required thickness of the near-wall layer of stretched cells.

Next, consider concave and convex ridges of the body surface. The stretched meshing near such ridges is usually of no concern when robustness is considered. However, it leads to lack of accuracy and poor convergence of the solver. To remedy this problem we will apply the similar approach, namely cluster the surface mesh to the ridges. The only difference from the corner treatment is that for ridges we will generate stretched surface elements. This technique is applicable for both concave and convex ridges.

Our surface generation technique is based on a CAD surface model. We can treat Coon's and trimmed NURBS surfaces as patches for the surface grid generator.

There are two important parameters characterizing the viscous mesh generation, namely the thickness of the near-wall layer of cells $\delta_{\min}$, and the factor q, which determines thickness increase from one layer to the next. The thickness of the grid cells increases in geometrical progression away from the wetted surfaces. As the first step in the grid generation process, all the sharp ridges of the wetted surfaces are identified and marked by measuring angles between the adjacent surfaces. Corner points are identified as points where more than two ridges meet.

The next step is the meshing of the trimmed CAD curves. If a curve starts or ends with a corner point, its discretization should be clustered to the corner point using the same geometrical progression law as will be used later in semi-structured grid generator. Thus, points next to the corner will be placed at the distance $\delta_{\min}$ apart from the corner. Curves, which are not identified as ridges, but are connected to ridges, are discretized in the similar way, namely, clustered to the points of their intersection with the ridges.

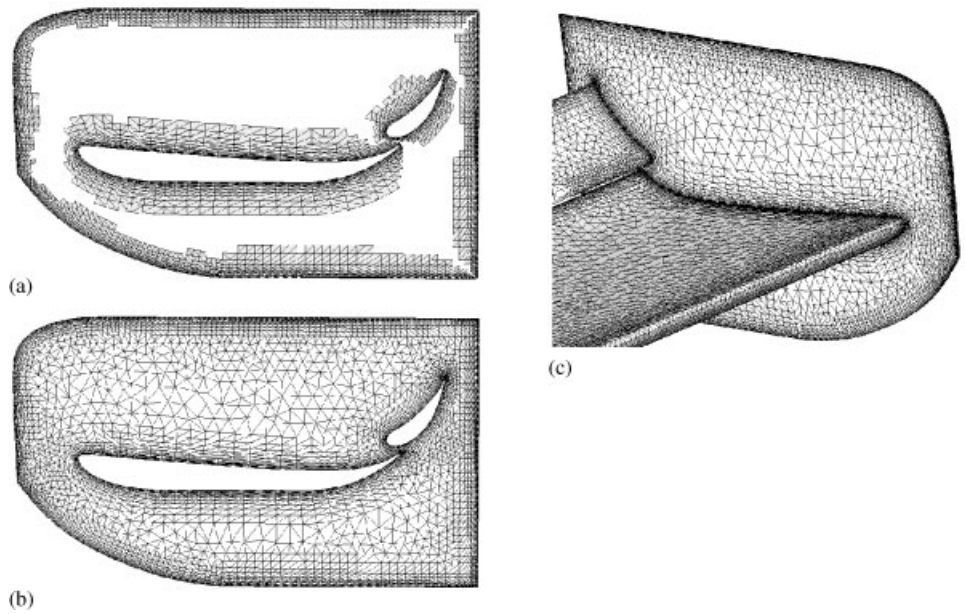

Figure 2. Surface meshing: (a) semi-structured surface mesh; (b) semi-structured and unstructured surface mesh; (c) assembly of the surface mesh.

Next, the surface patches are meshed. All ridges are used as an initial front for the normal point placement. Normals are computed at the ridges, and smoothing operations are performed on the normals. The smoothing is not performed for the corners so that the weight of the corner normals in the smoothing procedure is zero. This smoothing results in a structured-like mesh at concave corners, and in an isotropic triangular mesh at convex corners. Next, the layers of stretched rectangular mesh are generated along the normals starting from the ridges. The first layer of surface cells will have thickness $\delta_{\min}$, each subsequent layer being q times thicker than the previous one. Smoothing is performed for each layer. The advancement from a side stops if front overlapping occurs, if the cell size in normal direction exceeds its size in lateral direction, or if the thickness of the layer exceeds the local isotropic mesh size, which is prescribed by the background mesh. The resulting semi-structured mesh is shown for an example in Figure 2(a). As the viscous layers are generated, the regular triangular mesh is constructed in the rest of the surface domain. The sources are automatically placed into the corners so that the isotropic mesh at corners is clustered toward the source with the same geometric progression as the stretched grid is clustered towards the wetted surface. Examples of resulting surface meshes are shown in Figures 2(b) and 2(c). One can observe a smooth transition between the isotropic and stretched mesh at the corners. The isotropic advancing front triangulator meshes properly the gaps between the semi-structured layers. Figure 3 shows a close-up of the area with a problematic corner point. Since the vicinity of the corner point is meshed with isotropic triangles, generation of a volume mesh at this region should not pose a problem.

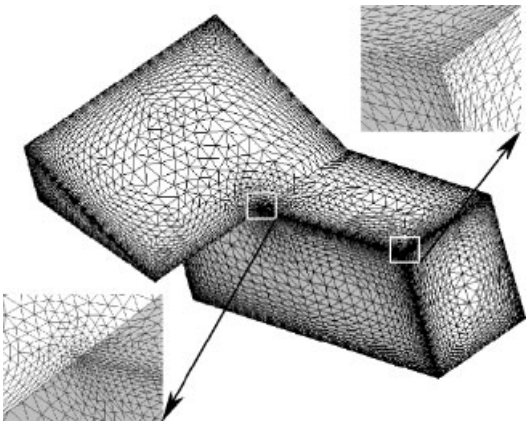

Figure 3. Surface meshing. Mesh at corners and ridges.

VOLUME GRID GENERATION ALGORITHM

After completion of the surface mesh, a semi-structured volume mesh is then generated, using a similar procedure. The normals at surface grid points are computed, and smoothing is applied to the normals. Smoothing is not applied at corners and concave ridges. In contrast, the normals at convex ridges are smoothed. For smooth surface regions, the standard semi-structured grid generation procedure [3] is employed. The edges of the surface triangulation are marked in such a way that all generated prisms can be subdivided into proper tetrahedra, in this way avoiding so-called Schönardt polyhedra. The semi-structured grid is not generated for corner points and concave ridges, since the isotropic tetrahedral grid generator is supposed to take care of these regions. Figure 4 shows the inflated surface after the semi-structured layers are generated. No mesh was generated in the vicinity of the corner points, but since the surface mesh was clustered to the corners, the exposed surface is composed of the isotropic triangles, which can serve as a good initial front for the isotropic tetrahedral generator.

The isotropic tetrahedra are generated using the advancing front method [10]. The volume mesh generator uses the sources, which were automatically defined at corner points, and some additional sources that are added at concave ridges. The sources at the corners are responsible for proper clustering toward the corners using the same geometrical progression law as for the stretched mesh generation. The ridge sources are automatically added to the concave ridges and contain stretching along the ridge's direction. These sources are designed to properly mesh narrow gaps between the viscous layers, which may occur in vicinity of the concave ridges. Some slices of the volume mesh, which corresponds to the surface mesh shown in Figure 3 are shown in Figure 5. The slice shown in Figure 5(a) is taken across the convex ridge, while the slice shown in Figure 5(c) is taken near the corner. Different shadings are used here to distinguish the cells obtained using the semi-structured grid generator from the cells generated by the isotropic tetrahedral grid generator. Figure 6 shows an internal mesh generated for the same geometry. Figure 6(a) shows a slice of the mesh taken near the corner.

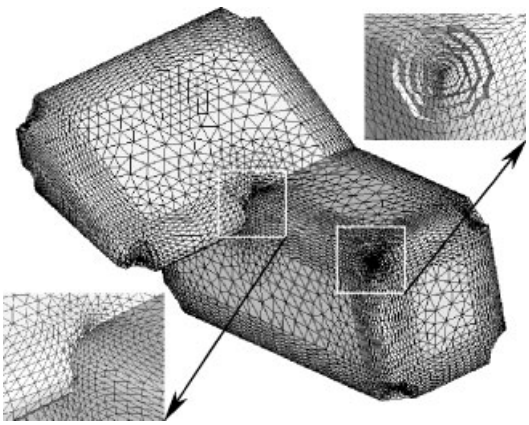

Figure 4. Volume meshing. Inflated surface of semi-structured mesh.

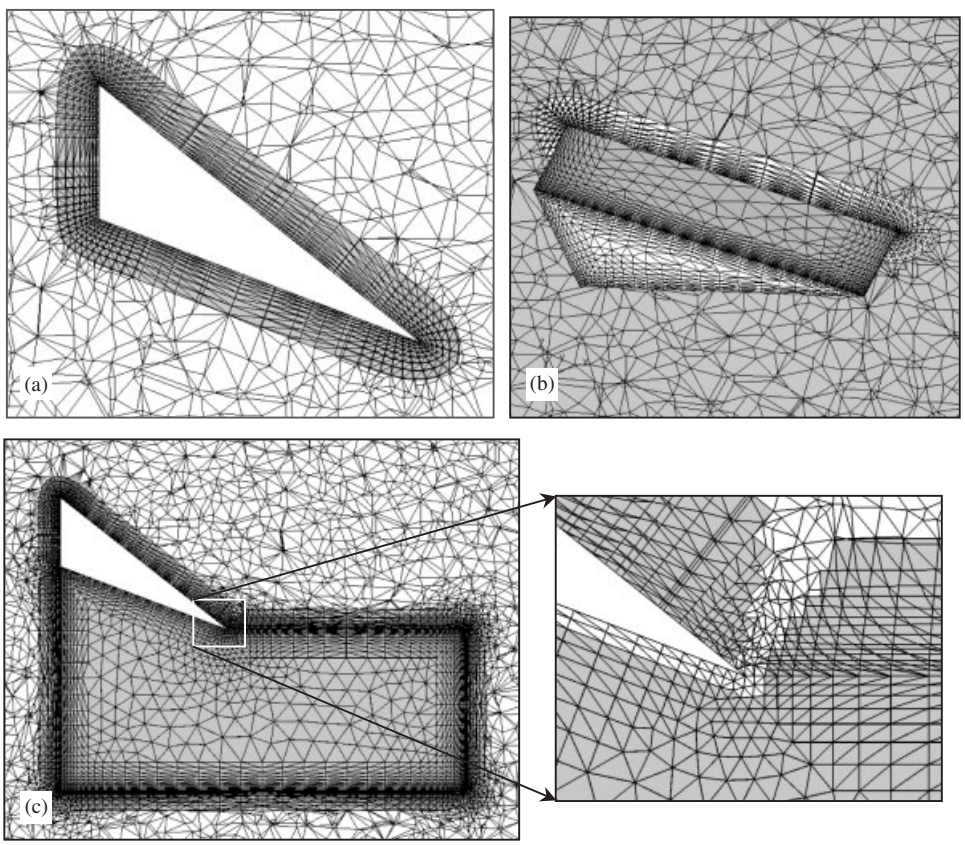

Figure 5. Volume mesh generation: (a) slice of external mesh; (b) slice of external mesh and part of surface mesh; (c) slice of external mesh near corner with close-up of the corner area.

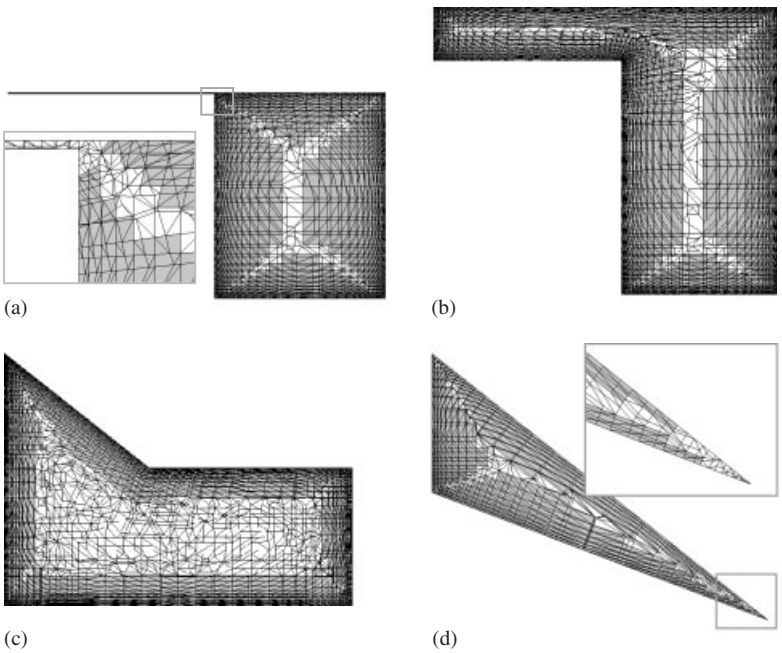

Figure 6. Volume mesh generation. Slices of internal mesh.

The slice shown in Figure 6(d) is taken across the concave ridge. One can readily observe the uniform resolution of all corners and ridges. It is important to note that while corner points are meshed using the isotropic tetrahedra, their size matches the thickness of stretched mesh layer for the given distance from the wall. This results in very smooth transition between the structured and unstructured grids, even for very complex geometries.

PARALLEL IMPLEMENTATION

The grid generation method was parallelized for cache-based shared memory computers. The parallelization method originally developed for advancing front unstructured mesh generators, described in Reference [10] was used.

EXAMPLES

ONERA M6

Mesh past an ONERA M6 wing has been generated using the proposed method. The mesh contains 4 744 776 tetrahedra. Figure 7(a) shows the inflated surface mesh in the vicinity of the trailing edge tip. The trailing edge tip was marked as a corner, so the inflated surface was not generated in the area. The visualization cut was made across the wing, so that the mesh around

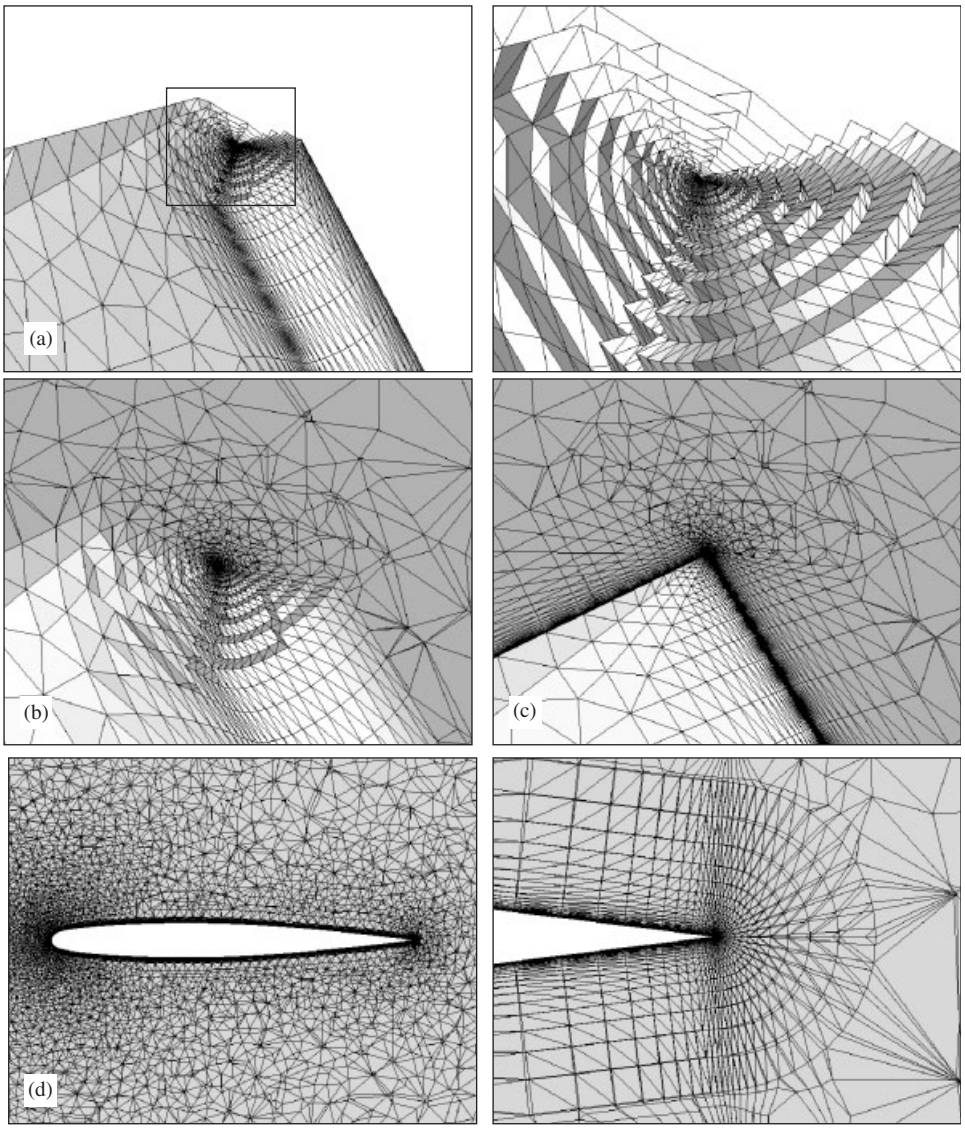

Figure 7. Mesh for ONERA M6 wing: (a) inflated surface at tip of trailing edge with close-up; (b) inflated surface and slice of tetrahedral mesh; (c) surface mesh and slice of tetrahedral mesh; (d) section of wing with trailing edge close-up.

the trailing edge can be seen. Figure 7(b) shows the inflated surface together with a cut of the volume mesh. Figure 7(c) shows a cut of the mesh and the wing surface mesh. Figure 7(d) shows a cut across the wing with a close-up at the trailing edge region. Even though we do not construct multiple normals at the trailing edge, the mesh is smooth in this region because of surface mesh clustering toward the trailing edge. The mesh provides very accurate transition

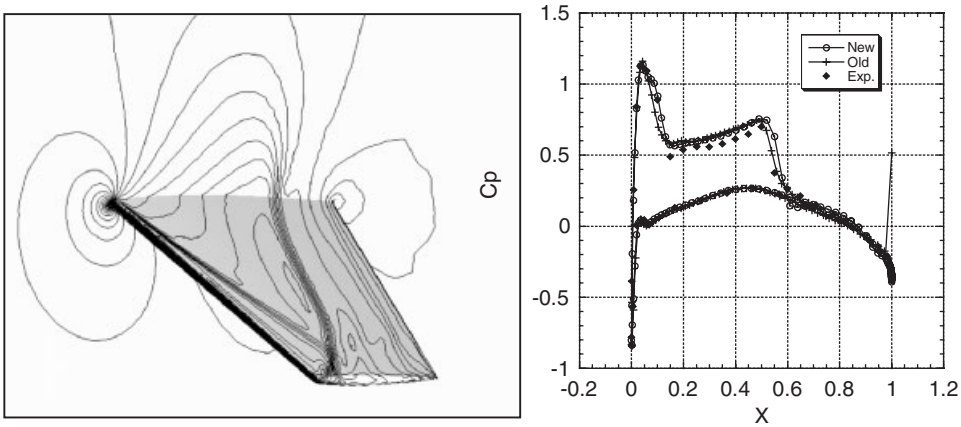

Figure 8. ONERA M6 wing, $M = 0.84$, angle of attack 3.06°, $Re = 11.72 \times 10^6$: (a) pressure contours and (b) Cp profile at 44% semi-span.

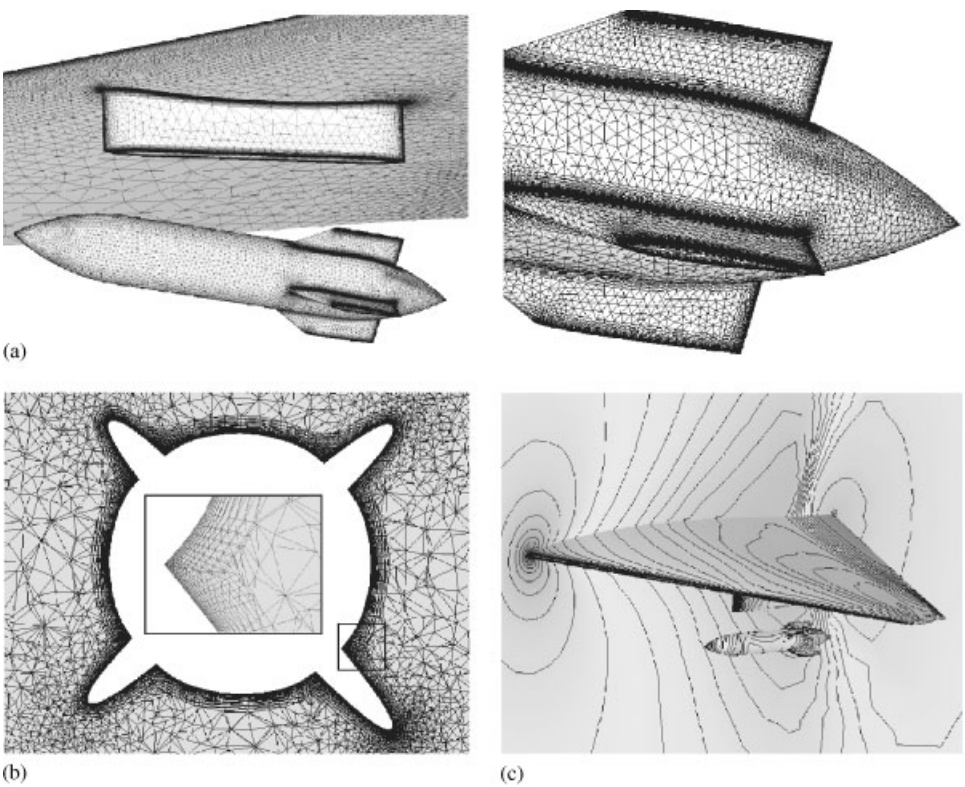

Figure 9. Wing/pylon/finned store configuration: (a) surface mesh fragments; (b) cut across the store with close-up of store/fin junction and (c) computed pressure contours.

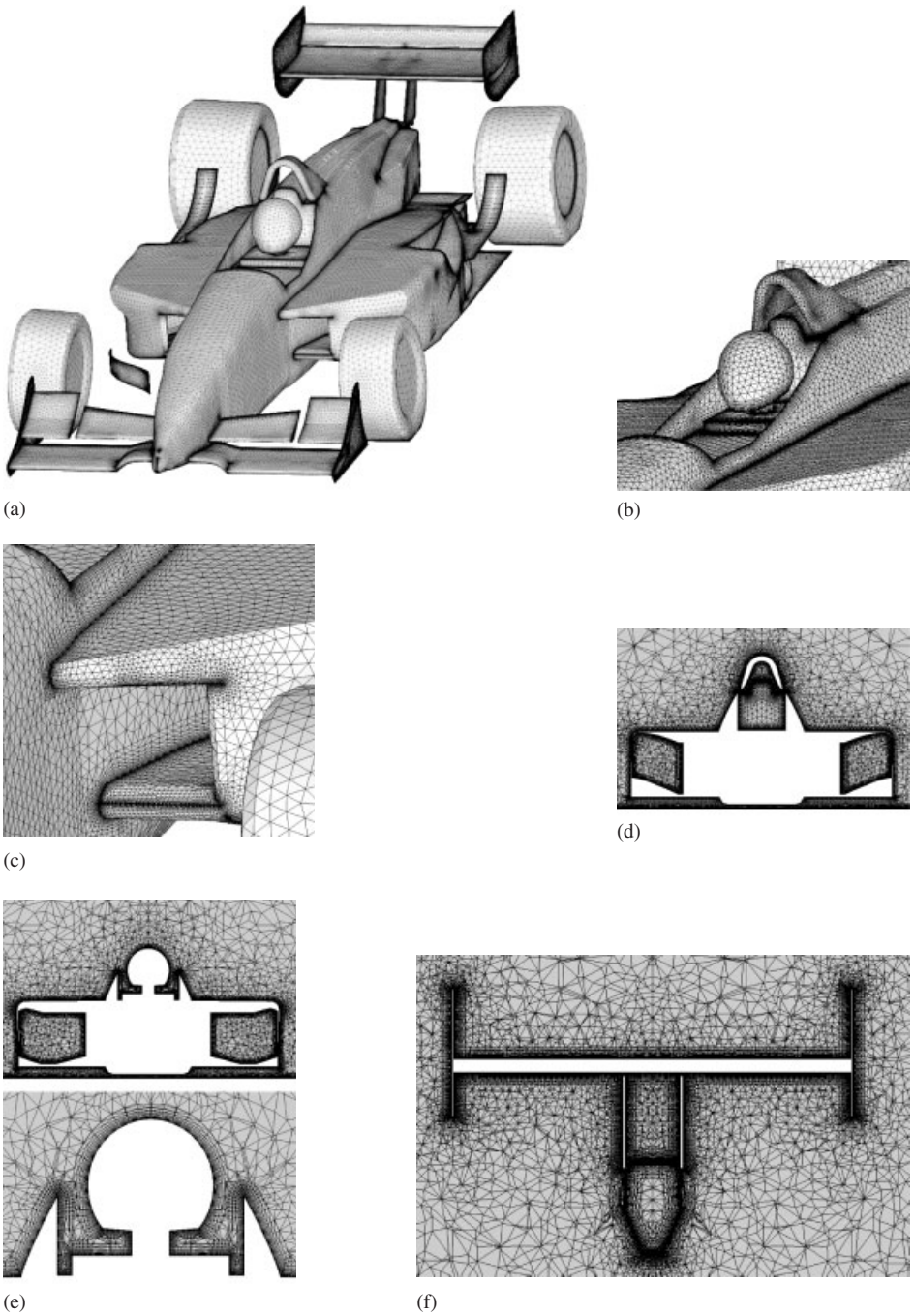

Figure 10. Generic racecar: Surface mesh, details of the surface mesh and planar cuts.

from the upper to the lower part of the wing in terms of the mesh size. The flowfield was computed with a Mach number $M = 0.8395$, Reynolds number $Re = 11.72 \times 10^6$, and an angle of attack of 3.06°. The Spallart-Allmaras turbulence model was used here. The solution was obtained using the containment dual finite volume upwind method with GMRES+LU-SGS [11, 12] implicit time integration. The pressure contours along the wing surface are shown in Figure 8(a). The *Cp* plot at the 44% section of the wing is shown in Figure 8(b). The result is in good agreement with experimental data. This result has been compared with results from another unstructured grid computation, conducted using an old grid generation approach (without the surface grid clustering to the ridges). The new mesh yields a much better solution in the vicinity of the trailing edge while the old mesh yields a non-physical solution with a huge *Cp* jump at the trailing edge.

Eglin test case

A wing/pylon/finned store configuration (the Eglin test case) is the next example. The geometry consists of a clipped delta wing, 45° sweep, composed of a constant NACA-64010 symmetric airfoil section. The wing has a root chord of 16 in, a semi-span of 13 in, and a taper ratio of 0.134. The pylon is located at the mid-span station. The width of the pylon is 0.294 in. A constant NACA-0008 airfoil section with a leading-edge sweep of 45° and a truncated tip defines the four fins of the store. The surface triangulation is shown in Figure 9(a). The volume mesh consists of 6 719 666 tetrahedra. A cut across the store is shown in Figure 9(b). The flow solutions presented for Mach number of 0.95 and an angle of attack of 0°. Figure 9(c) shows the pressure contours on the wing/store surface.

Racecar

A mesh for a generic Indy-type racecar geometry is shown in Figure 10. The overall surface mesh is shown in Figure 10(a). Figures 10(b) and 10(c) focus on particular regions of the mesh (pilot compartment and air intake). Slices of the volume mesh across the car's spoiler and pilot compartment are shown in Figures 10(d)–10(f). The mesh contains 6 074 832 tetrahedra.

It should be noticed that even though the mesh clustering at corners and ridges results in larger size of mesh, the robustness and mesh quality gains are much more important even if we have to pay by doubling the mesh size.

CONCLUSIONS

A method for automatic generation of Navier–Stokes meshes is proposed. The method is designed to avoid pitfalls associated with poor mesh quality or failures in the vicinity of sharp ridges and corners. The proposed solution is based on surface mesh clustering toward the corners. The examples presented demonstrate the efficiency of the proposed technique for very complex geometries. The resulting meshes yield more accurate Navier–Stokes solutions.

REFERENCES

1. Nakahashi K. Marching grid generation for external viscous flow problems. *Transactions of the Japan Society for Aeronautical and Space Sciences* 1992; **35**(108):88–102.
2. Sharov D, Nakahashi K. Hybrid prismatic/tetrahedral grid generation for viscous flow applications. *AIAA Journal* 1998; **36**(2):157–162.
3. Löhner R. Matching semi-structured and unstructured grids for Navier–Stokes calculations. *Proceedings of the AIAA 11th CFD Conference*, AIAA-93-3348-CP, 1993; 555–564.
4. Connell SD, Braaten ME. Semi-structured mesh generation for 3-D Navier–Stokes calculations. *AIAA* 1995; 95–1679.
5. Prizadeh S. Unstructured viscous grid generation by the advancing layers method. *AIAA Journal* 1994; **32**(8):1735–1737.
6. Garimella R, Shepard MS, Webster BE. Automatic mesh generation of complex configuration including viscous boundary layers. *10th International Conference on Finite Elements in Fluids* 1998; 369–374.
7. Wang ZJ. An automated viscous adaptive cartesian grid generation method for complex geometries. *6th International Conference On Numerical Grid Generation in Computational Field Simulations* 1998; 577–586.
8. Löhner R. Generation of unstructured grids suitable for RANS calculation. *Proceedings of the AIAA 14th CFD Conference*, AIAA-99-3251-CP, 1999; 9–18.
9. Liou M-S, Zheng Y. A Three-Dimensional Hybrid Grid: DRAGON Grid, 1st ICCFD, 2000.
10. Löhner R. A Parallel Advancing Front Grid Generation Scheme, AIAA-00-1005, 2000.
11. Luo H, Baum JD, Löhner R. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics* 1998; **146**:664–690.
12. Luo H, Sharov D, Baum JD, Löhner R. On the computation of compressible turbulent flows on unstructured grids. AIAA-2000-0926, 2000.

